Statistical attribute filtering to detect faint extended astronomical sources

Teeninga, Paul; Moschini, Ugo; Trager, Scott C.; Wilkinson, Michael H. F.

Published in:
Mathematical Morphology - Theory and Applications

DOI:
10.1515/mathm-2016-0006

Publication date:
2016

Research Article **Open Access**

Paul Teeninga, Ugo Moschini*, Scott C. Trager, and Michael H.F. Wilkinson

Statistical attribute filtering to detect faint extended astronomical sources

DOI 10.1515/mathm-2016-0006
Received July 9, 2015; accepted February 18, 2016

Abstract: In astronomy, sky surveys contain a large number of light-emitting sources, often with intensities close to the noise level. Automatic extraction of astronomical objects is therefore needed. SExtractor is a widely used program for automated source extraction and cataloguing, but it is not optimal with faint extended sources. Using SExtractor as a reference, the paper describes an improvement of a previous method proposed by the authors. It is a Max-Tree-based method for extraction of faint extended sources without using a stronger image smoothing. The Max-Tree structure is a hierarchical representation of an image, in which attributes can be computed in every node. Object detection is performed on the nodes of the tree and it relies on the distribution of a statistic calculated using the power attribute, compared to the expected distribution in case of noise. Statistical tests are presented, a comparison with the object extraction of SExtractor is shown and results are discussed.

Keywords: Attribute filters, statistical tests, astronomical imaging, object detection

1 Introduction

In astronomy, sky surveys contain a huge quantity of light-emitting sources, representing astronomical objects. With advances in technology, an increasing number of images and volumes at both high resolution and high bit-depths becomes available. Manually extracting every object is not feasible, due also to the low intensities of many sources, often close to the noise level. Object detection can be seen as the process of separating groups of pixels that belong to a source from those that belong to noise or background. Masias et al. [7] presented a detailed overview of state-of-the-art object detection techniques in astronomy. The authors stress the fact that many astronomical objects do not show clear boundaries and have intensities close to the detection level of the instrument. Besides, the size of relevant objects in an image can vary greatly. To detect sources in astronomical images, two main categories of methods are prominent: thresholding and local peak search. With the former method, connected sets of pixels are considered an object if they are above a certain threshold value; with the latter, objects are identified descending to lower intensities from the pixels representing image maxima. In recent years, methods based on component trees or max-trees have been used to process greyscale or mono-channel images. They rely on a hierarchical representation of an image, finding connected sets of pixels at every intensity level. The tree structure can be augmented with attributes related to every node for image filtering or segmentation purposes. Such structures have been already successfully used for astronomical object detection [2, 10, 11]. Source Extractor (SExtractor) [3] is a state-of-the-art software for automatic extraction of astronomical objects. It is based on thresholding and it works on many data types, such as optical, infra-red and radio datasets. For the purpose of this work, we used a dataset extracted from the Sloan Digital Sky Survey [16] (SDSS) Data Release 7 [1] catalogue, containing optical images. The whole cata-

***Corresponding Author: Ugo Moschini:** Johann Bernoulli Institute, E-mail: u.moschini@rug.nl
Paul Teeninga: Johann Bernoulli Institute, E-mail: p.teeninga@home.nl
Michael H.F. Wilkinson: Johann Bernoulli Institute, E-mail: m.h.f.wilkinson@rug.nl
Scott C. Trager: Kapteyn Astronomical Institute, E-mail: sctrager@astro.rug.nl

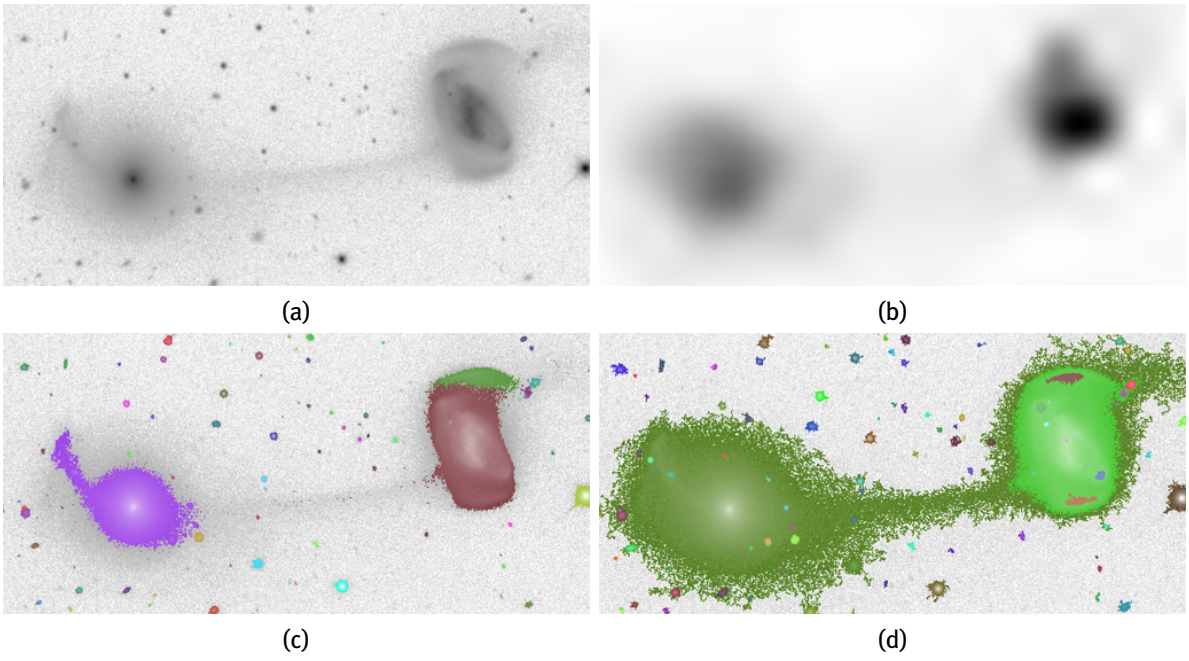
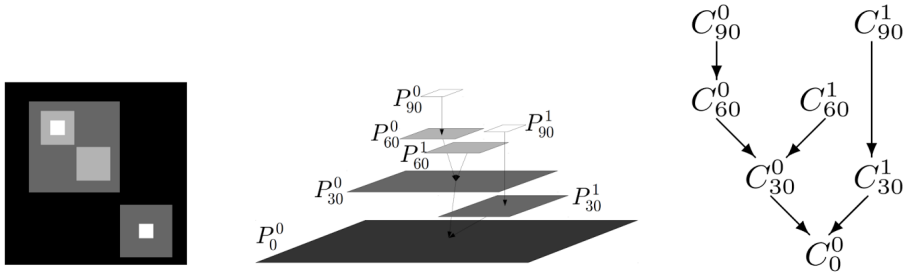


Figure 1: (a) original cropped section from the file `fpC-002078-r1-0157.fit` of the SDSS DR7 catalogue, showing two interacting galaxies; (b) the background estimate of SExtractor shows correlation with the objects; (c) result of the segmentation by SExtractor with default settings and (d) by the proposed method. The filament between the galaxies is not extracted in (c).

logue contains 357 million unique objects, representing a perfect example of the reason why automatic object detection is needed. Qualitative and quantitative comparisons between SExtractor and other methods can be found in [7, 8]. Its main disadvantages are two: the selection of the optimal threshold above which pixels are considered as an object and the detection of the fainter structures, often faulty. SExtractor first estimates the image background. An image background, caused by light produced and reflected in earth's atmosphere, is estimated and subtracted before thresholding. In the SDSS data set, SExtractor's estimate shows bias from objects (see Fig. 1), which reduces their intensities. With the default settings, to perform a correct segmentation and avoid false positives, objects are identified with the pixels with intensity at a threshold level higher than 1.5 times the standard deviation of the background estimate at that location. We refer here to such mechanism as *fixed threshold*: the threshold value relies only on local background estimates in different sections of the image and it ignores the actual object properties. To identify nested objects, larger regions are later deblended, re-thresholding at 32 quantized levels, logarithmically spaced between the threshold value and the peak intensity in the region. Deblending occurs when the integrated intensity is above a certain fraction of the total intensity and if another branch with such property exists.

In this paper, we propose a solution to improve the fixed threshold approach and the quantized deblending step. For our experiments, we selected a subset of 254 images from the SDSS DR 7, containing mergers and overlapping galaxies. Merging galaxies often show faint extended structures due to their interaction and the tidal forces between them. Overlapping galaxies look close to each other at the same location in the sky, but they are not interacting and they might be in fact very distant. Our own background estimate, introduced in [17] and more extensively explained in [18], returns a constant value of the background: on the SDSS dataset, the object bias present in the estimate of SExtractor is reduced. After a software bias is subtracted from the images, the pixel values are proportional to photo-electron counts [13]. The distribution of background pixels is approximately independent Gaussian, with a variance which varies linearly with intensity. In our detection method, the supporting data structure is a Max-Tree [12] created from the image, where every node corresponds to a connected component for all the threshold levels in the original image. The choice was inspired by the simplified component tree used in the SExtractor deblending step and was already suggested in [11]. Here we extend our solution proposed in [17], in which a Max-Tree based method varies locally



(a) 2D image (b) Peak components (c) Max-Tree

Figure 2: (a) a grey-scale 2D image with intensities from 0 to 90, (b) its peak components P^k_h at intensity h and (c) the corresponding Max-Tree nodes C^k_h.

the threshold depending on object size by using a statistical test rather than arbitrary thresholds on the attributes computed in the nodes of the tree. The distribution of an attribute, the power [19], is studied with respect to its expected distribution in case of noise components. Nodes are marked significant if noise is an unlikely cause, for a given significance level. The significance level is an intuitive parameter, identified with the likelihood of marking a node as significant. We present an extension of our method [17] by analysing variations of the attributes used and giving a more extended explanation and discussion of the results. In Section 2 and 3, we introduce briefly our background estimation and the Max-Tree structure. In Section 4, statistical tests to separate noise from objects, based on the distribution of the power in case of noise, are discussed. In Section 5, object detection and deblending are explained and in Section 6 a comparison with SExtractor is presented, followed in the next sections by conclusions and future directions of research.

2 Max-tree structure

Any grey-scale image can be represented as a set of connected components, that are groups of pixels pathwise connected and with the same intensity, according to the classical definition of connectivity among pixels in [14]. An image can be thresholded at every intensity level and the connected (peak) components are identified, at each level. Since the intensities can be ordered, peak components can be nested one on top of the other. Such inclusion relationship among components is translated into a hierarchical structure: the Max-Tree [12]. Every node in the tree corresponds to a peak component. The root of the tree corresponds to the entire image, while the leaves represent the local maxima of the image. Nowadays, many algorithms can build efficiently max-trees of images that carry high bit-depth integers and floating point values, often found in astronomical data. Fig. 2b illustrates the hierarchy of components at different intensities h for the image in Fig. 2a. Fig. 2c shows the Max-Tree corresponding to the peak components. The root component is the black background and the two leaves correspond to the image maxima. The arrows represent parent-child relationships. Useful attributes related to the components can be computed in the node while the Max-Tree is being built. The attributes are used in the filtering stage to choose which nodes must be preserved. This process is referred to as connected *attribute filtering* [4]. The simplest example of filtering is to preserve the components with area larger than a given threshold: the tree is parsed and nodes whose area attribute does not satisfy the threshold value are not considered. Meaningful attributes allow for selecting components for a given purpose. For example, Perret et al. [11] and Berger et al. [2] defined attributes for object detection on multi-spectral data and optical data, respectively. Once components are selected, an output image is created by parsing the tree and visualizing only the nodes preserved according to the threshold value. Several rules define the new intensity to be assigned to the pixels corresponding to the nodes that have been filtered out.

Algorithm 1: `IsFlat`(T, α) [18]

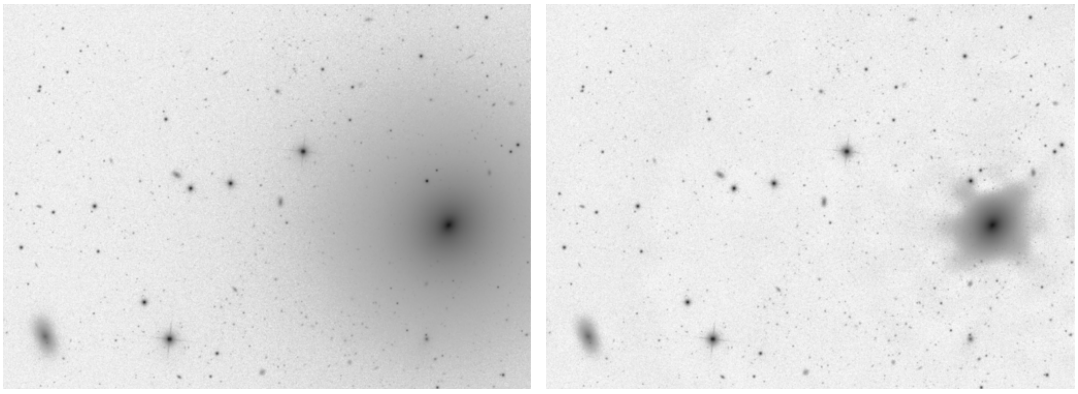
Input: $w \times w$ tile T, rejection rate α.
Result: True if T is flat. False otherwise.

1 $\alpha_1 \leftarrow 1 - (1-\alpha)^{1/2}$;
2 Perform the D'Agostino-Pearson K^2-test on the values of T with rejection rate α_1. Return false if rejected;
3 $(T_{1,1}, T_{1,2}, T_{2,1}, T_{2,2}) \leftarrow \frac{w}{2} \times \frac{w}{2}$ tiles partition of T.;
4 $\alpha_2 \leftarrow 1 - (1-\alpha)^{1/4}$;
5 Perform a t-test of equal means on the pairs $(T_{1,1} \cup T_{1,2}, T_{2,1} \cup T_{2,2})$ and $(T_{1,1} \cup T_{2,1}, T_{1,2} \cup T_{2,2})$ using rejection rate α_2. Return false if the null hypothesis of equal means (and variances), in any of the two tests, is rejected;
6 Return true;

3 Background estimation

A more detailed description of the algorithm used to estimate the background in the SDSS DR7 dataset was proposed in [17] and explained more extensively in [18]. However, in this section, we report a brief overview of our background estimation method that will help to give a better understanding of the object detection method. For the SDSS dataset, an image is assumed to be the sum of a background image B, noiseless image O and Gaussian noise. Actually, Poissonian noise would dominate, but due to the high photon counts already at the minimum intensity, the distribution is approximately Gaussian, with a variance which varies linearly with the image intensity. The noise variance is equal to $g^{-1}(B+O)+R$, with g equivalent to the CCD *gain* and R due to other noise sources, such as read noise, dark current and quantisation. The background is approximated by the mean value of flat tiles. These are regions in the image devoid of objects. It is a constant estimate for the whole image. When objects are not present in a tile, its pixel values should have been drawn from a Gaussian distribution, given our noise model. Two statistical tests are applied to the tiles. In first instance, a normality test using the D'Agostino-Pearson K^2-statistic [5] is used to select flat tiles candidates. Then, t-tests of equal means in different parts of the tile are used, because the normality test alone does not consider the location of pixel values: tiles with a near-linear slope due to objects could be wrongly considered as flat. More details on the choice of the size of a tile and the rejection rate used in the previous tests are in [18]. The pseudo-code that checks if a tile is flat is reported in Alg. 1. Flat tiles of size 64x64 pixels are chosen. Every image contains at least one. Smaller sizes are not ideal, because the estimate has an higher chance to be biased by the presence of astronomical objects.

By contrast, the background estimate of SExtractor is not constant but adaptive: it often shows correlations with larger objects. An example can be seen in Fig. 1b: the strong correlation with the disks of the two galaxies is evident. It is then more difficult to detect faint parts of the objects, because they could be considered background and deleted after the background is removed from the image, prior to the object segmentation. As an example, the thin structure between the merging galaxies in Fig. 1d is better preserved than in Fig. 1c: the intensity of the background estimated by our method is lower than the intensity of the faint interconnecting filament. Another issue with the SExtractor's estimate is that the problem of shape distortion of objects always appears in case of non-constant estimates, as Fig. 3 illustrates. In [18], it is shown that a constant estimate for the background is a suitable choice for the SDSS dataset. In the hypothesis that the background is not flat, a fit closer to the local estimates should be better. However, as it was seen experimentally, that would increase segmentation errors at the locations that correlate with objects. Inspecting images where the distance between the estimate and the expected distribution of the mean background value was high showed that this is due to the presence of large galaxies and not to changes in the background intensity.



(a) (b)

Figure 3: [18] (a) image with a constant background estimate subtracted; (b) image with the SExtractor background estimate subtracted. Its shape is distorted by the background correlations. SDSS file used is `fpC-003836-r4-0249.fit`.

With the background removed, the variance of the noise is $g^{-1}O + \sigma_{\text{bg}}^2$, where $\sigma_{\text{bg}}^2 = g^{-1}B + R$ is what our estimate represents. Negative image values after background subtraction are set to 0 and the Max-Tree is built. The next step is to identify nodes that are part of objects, referred to as *significant* nodes.

4 Identifying significant nodes

Four significance tests are defined in this section. Their aim is to mark a node as significant if one or more objects are represented by the pixels of the node, given our background estimate. To identify the nodes in the tree belonging to objects, we will start from the definition of the power [19] attribute. It is a measure similar to the definition of object *flux*, often used in astronomy, or the integrated intensity, used by SExtractor. Let us define the intensity associated with a node P with $f(P)$. Similarly, let $f(x)$ be the value of a pixel x. Let us define also P_{anc} as the closest significant ancestor of a component P. If no such node exists, P_{anc} is equal to the root node. P_{anc} also represents the local background of a component. P is significant if it can be shown that $\exists x \in P : O(x) > f(P_{anc})$, for a given significance level α. We will use two different definitions of the power attribute of P:

$$\text{power}(P) \quad := \quad \sum_{x \in P} (f(x) - f(\text{parent}(P)))^2 \tag{1}$$

and

$$\text{powerAlt}(P) \quad := \quad \sum_{x \in P} (f(x) - f(P_{\text{anc}}))^2. \tag{2}$$

To determine if a node P is due to noise or not, we will study the distribution of the power values for different components' areas, with respect to its expected distribution in case of noise nodes. Noise scales linearly with the intensity level. To filter local maxima due to noise on top of objects, the local background of an object can be higher than our constant estimate. Therefore, to normalize the power attribute for the nodes that do not have the root (background) as parent, the power is divided by the local background variance $\sigma^2 = \hat{\sigma}_{\text{bg}}^2 + g^{-1} \cdot f(parent)$. The parent component, or the closest significant ancestor in some cases, are considered as local background. To identify significant nodes, four significance tests that use the two attributes above are defined in the following.

4.1 Significance test 1: power given area of the node.

A node P is considered significant it is possible to provide a statistical test to show that $O(x) > f(P_{\text{anc}})$ for pixel locations $x \in P$, given a significance level α. We use the following hypothesis:

$$H_{\texttt{power}} := \forall x \in P : O(x) \leq f(\text{parent}(P)).$$

This test uses the definition of power attribute in Equation 1. In the limit case, $\forall x \in P : O(x) = f(\text{parent}(P))$ for pixels $x \in \text{parent}(P)$. In this test, we assume that the distribution of the power attribute scaled by the variance σ^2 for noise nodes follows a χ^2 distribution. In fact, in the case of Gaussian noise, the power of noise components is a sum of squared independent variables and therefore it follows such distribution. For a random pixel x in P, the value $(f(x) - f(\text{parent}(P)))^2/\sigma^2$ has a χ^2 distribution with 1 degree of freedom. If P is due to noise, it has a χ^2 distribution with degrees of freedom equalling to the area of P. Let us define a function $\texttt{inverse}\chi^2\texttt{CDF}(\alpha, \texttt{area})$ that returns the rejection boundary given by the χ^2 cumulative distribution function (CDF), for a significance level α. The χ^2 CDF (or inverse) is commonly available in scientific libraries. An example of a rejection boundary of a χ^2 CDF is shown in Fig. 4a. If $\texttt{power}(P)/\sigma^2 > \texttt{inverse}\chi^2\texttt{CDF}(\alpha, \texttt{area})$, $H_{\texttt{power}}$ is rejected: $O(x) > f(\text{parent}(P)) \geq f(P_{\text{anc}})$, for some pixels $x \in \text{parent}(P)$, making P significant.

A precise χ^2 distribution of the power attribute holds for the nodes that have the root as parent. For the other nodes, the rejection boundary is a conservative model and minimizes the number of false positives. In significance test 1, leaf nodes are less likely to be found significant due to their small area and the low intensity difference with the parent node. Some nodes could be erroneously marked as noise even if they are not. The next three tests use the alternative definition of the power attribute in Equation 2 to address this issue: the power attribute has larger values than in Equation 1.

4.2 Simulating distributions

In all the next three significance tests (all right tailed) the exact distribution of powerAlt is not known and it is obtained by Monte Carlo simulation. Gaussian noise images are generated, with mean and variance equal to our estimates. A number n of independent values is generated. On average, given a significance level α, the number of false positive equals to $r = \alpha \cdot n$ nodes: the attribute of the false positive nodes is greater than or equal to the rejection boundary. The best estimate of the rejection boundary, without any further information about the distribution, is the average on many noise images of the two smallest of the $r + 1$ largest values of the attribute.

4.3 Significance test 2: powerAlt given area and distance.

In this test, we use the following hypothesis:

$$H_{\texttt{powerAlt}} := \forall x \in P : O(x) \leq f(P_{\text{anc}}).$$

To make the significance level more constant for every node, independently of its height in the tree, we refer to its ancestor rather than to the parent node in the computation of the power attribute. The definition of power attribute in Equation 2 is used. Let us assume $H_{\texttt{powerAlt}}$ is true and consider the extreme case $\forall x \in P : O(x) = f(P_{\text{anc}})$. Let us define $\texttt{distance}(P) := f(P) - f(P_{\text{anc}})$. Let X be a random set of area(P) - 1 values drawn from a truncated normal distribution with a minimum value of $\texttt{distance}(P)$. The variance is set to $\sigma^2 = \hat{\sigma}^2_{\text{bg}} + g^{-1} f(P_{\text{anc}})$. Attribute $\texttt{powerAlt}(P)$ has the same distribution as $\texttt{distance}^2(P)$ plus the sum of the squared values in X. Let the function $\texttt{inversePowerAltCDF}(\alpha, \texttt{area}, d)$) return the estimated rejection boundary for the power attribute, for given α, area and distance values. Hypothesis $H_{\texttt{powerAlt}}$ is rejected if $\texttt{powerAlt}(P)/\sigma^2 > \texttt{inversePowerAltCDF}(\alpha, \texttt{area}, d)$): it means that the object image $O(x)$ at some pixels x in P is higher than $f(P_{\text{anc}})$, and P is marked as significant. The minimum area of a significant node is 2 pixels.

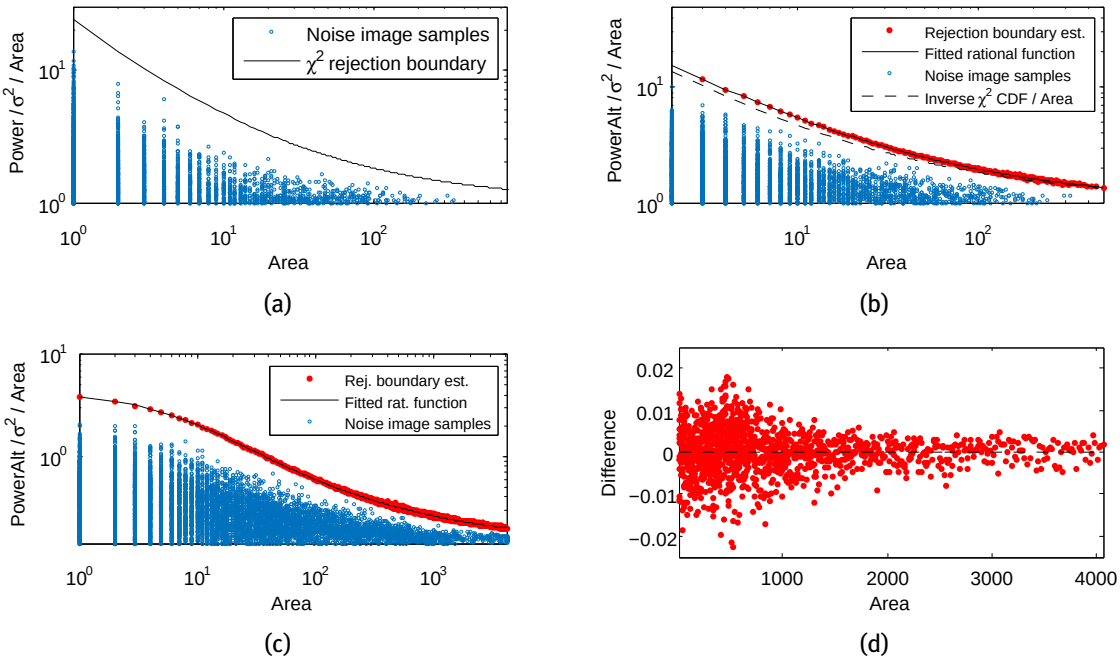


Figure 4: (a) rejection boundaries for significance test 1 and (b) the simulated rejection boundaries for test 3 and (c) test 4, log-log scaled; (d) shows the difference between the rational function and its estimate in (c). $\alpha = 10^{-6}$.

An estimate is given for `inversePowerAltCDF` for constant α, varying `area` and `distance`. Random samples are generated given several values of area and distance: the range for the area values goes from 2 to 768 pixels, while distance has a maximum value of 4, with 0.25 as step size. For each rejection boundary, varying `distance`, a rational function is fitted to reduce the error and the storage space. In the tests, the rational functions appear to be valid approximations. Fig. 5a shows the difference between the rational function and the rejection boundary obtained for significance test 2. Fig. 5b shows the rational functions, with polynomials of degree 3. Let *rms* be the root mean square of the differences between a rejection boundary estimate and rational function. The maximum value of *rms* is 0.019. When it is not possible for the expected values of the estimates to be the same as the rejection boundary, the choice is made to prefer overestimation, as it will not increase the number of false positives. Linear interpolation between rejection boundaries is used if a boundary is not available for a `distance`, which happens in nearly all cases.

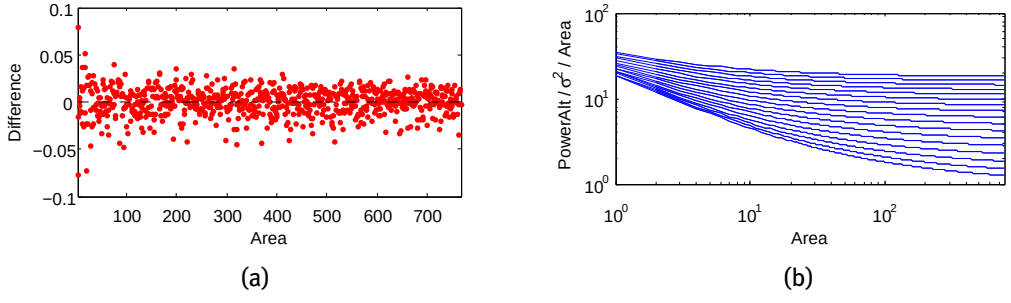


Figure 5: (a) shows the difference between the rational function and the simulated rejection boundary for `powerAlt` given `distance` = 0, $\alpha = 10^{-6}$, in significance test 2; (b) shows the approximated rejection boundaries for the `powerAlt` attribute: `distance` = 0 for the bottom curve and `distance` = 4 for the top curve with a step size equals 0.25 ($\alpha = 10^{-6}$, log-log scaled).

Algorithm 2: `SignificantNodes`$(M, \texttt{nodeTest}, \alpha, g, \hat{\sigma}^2_{\text{bg}})$

Input: Max-Tree M, significance test *nodeTest*, significance level α, gain g, variance of the background $\hat{\sigma}^2_B$.
Result: Nodes in M that are unlikely to be noise are marked as significant.

```
forall the nodes P in M with f(P) > 0 in non-decreasing order do
    if nodeTest(M, P, α, g, σ̂²_bg) is true then
        Mark P as significant;
```

4.4 Significance test 3: `powerAlt` given `area`.

Significance test 3 uses the distribution of `powerAlt` given α and area of a component. It is independent of the `distance` measure, not used as parameter in the inverse CDF. Using the assumptions from the significance test 2, `distance`(P) has a truncated normal distribution with a minimum value of 0, the same distribution as a random non-negative pixel value. The rejection boundary is calculated through simulated noise images. Fig. 4b shows the rejection boundary estimate and the fitted rational function for this significance test. Four-connectivity is used. A different connectivity would possibly change the rejection boundary.

4.5 Significance test 4: `powerAlt` given `area`, using a smoothing filter.

It is equal to significance test 3 with the only difference that the image is smoothed beforehand. Smoothing is used to reduce noise and to detect more objects. A larger number of objects is detected with this test. We use the same smoothing filter used in SExtractor:

$$H = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Filtering is done after background subtraction and before setting negative values to zero. After smoothing, pixel values are not independent any more. Decision boundaries are determined again through Monte Carlo simulations. Fig. 4c shows the rejection boundary with its fitted rational function and Fig. 4d shows the difference between the rational function approximation and the estimates for this significance test.

4.6 Testing the nodes

Alg. 2 describes the method used for marking nodes not due to noise as significant. Visiting nodes in non-decreasing order by pixel value simplifies the identification of P_{anc}, if stored for every node. In the case of significance test 1, nodes can be visited in arbitrary order. Function `nodeTest`$(M, P, \alpha, g, \hat{\sigma}^2_{\text{bg}})$ in Alg. 2 performs the significance test and returns true if P is significant, false otherwise.

4.7 Value of significance level α

The Max-Tree of a noise image after subtraction of the mean and truncation of negative values is expected to have $0.5n$ nodes, with n the number of pixels. An upper bound on the number of expected number of false positives is $\alpha \cdot 0.5n$ if the nodes are independent. Given a 1489×2048 noise image, the same size of the images in the data set, and $\alpha = 10^{-6}$, the upper bound on the expected number of false positives is approximately 1.52, given a right-tailed distribution. We performed a test on noise images and the actual number of false positives observed turned out to be lower. An estimate of the actual number of false positives is 0.41, 0.72, 0.94 and 0.35

Algorithm 3: `FindObjects`(M)

Input: Max-Tree M.
Result: Nodes in M that represent an object are marked.

```
forall the significant nodes P in M do
    if P has no significant ancestor then
        Mark P as object;
    else if mainBranch(P_anc) is not equal to P then
        Mark P as object;
```

for the four significance tests, respectively, averaged over 1000 simulated noise images. Argument α is set to 10^{-6} by default.

5 Finding objects

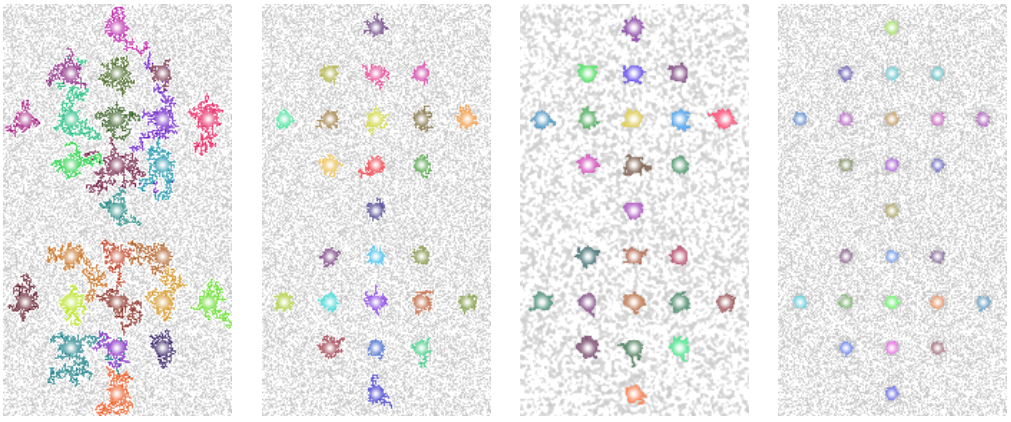
After that nodes have been marked as significant, it must be considered that multiple significant nodes could be part of the same object. A significant node with no significant ancestor is marked as an object. Let `mainBranch`(P) be the function returning a significant descendant of P with the largest area, as in Alg. 3. A significant node, with significant ancestor P_{anc}, that differs from the one returned by `mainBranch`(P_{anc}) is marked as a new object. This operation of identifying nested actual objects on top of a larger one is called deblending. The decision if a node is considered a new object depends on the used significance test, smoothing filter and connectivity, as it will be shown in the comparison section. The procedure of marking nodes as objects is summarised in Alg. 3.

5.1 Moving object markers up: parameter λ

Nodes marked as objects have a number of pixels attached due to noise. The number decreases at a further distance from the background signal. Object markers can be moved up in the tree, for λ times the standard deviation of the noise. The obvious choice for an object node P is `mainBranch`(P), if such a node exists, since it does not conflict with other object markers. Otherwise, the descendant of P with the highest p-value found with the corresponding CDF for its `power` or `powerAlt` attribute value would be the perfect candidate. However, the CDF is not always available or easy to store. Instead, the descendant with the largest `power` attribute is chosen, if at least one exists. The function that returns the descendant is called `mainPowerBranch`(P). Alg. 4 illustrates the method. An alternative to allowing a lower value of $f(P_{\text{final}})$ in Alg. 4 is to remove those object markers. If the parameter λ is set too low, there are too many noise pixels attached to objects. However, to be able to display faint parts of extended sources a low λ is preferred. We performed tests on objects simulated with the IRAF software, generating 25 stars with low magnitude (-5) and adding Gaussian noise at every location with the pixel value as mean and variance. If parameter λ is set too low, as in Fig. 6(a), there are too many noise pixels attached to objects. The object shapes in Fig. 6(d) look better. However, to be able to display faint parts of extended sources a low λ is preferred, therefore $\lambda = 0.5$ is used as a compromise. Experimentally, such value of λ worked effectively on the SDSS data set.

6 MTObjects vs Source Extractor

Alg. 5 summarises the whole procedure from background estimation to object identification. The proposed method is called MTObjects (Max-Tree Objects), since astronomical object detection is obtained using a Max-



(a) (b) (c) (d)

Figure 6: Twenty-five simulated stars, logarithmic grey scale: (a) $\lambda = 0$; (b) $\lambda = 0.5$; (c) $\lambda = 0.5$, with the image smoothed using the default SExtractor filter; (d) $\lambda = 2$.

Algorithm 4: `MoveUp`$(M, \lambda, g, \hat{\sigma}^2_{\text{bg}})$

Input: Max-Tree M, factor λ, gain g, variance of the noise at the background $\hat{\sigma}^2_{\text{bg}}$.
Result: For every object marker that starts in a node P and moves to the node P_{final}:
$f(P_{\text{final}}) \geq f(P_{\text{anc}}) + \lambda$ times the local standard deviation of the noise, when possible. $f(P_{\text{final}})$ might be lower if P_{final} has no descendants.

```
forall the nodes P in M marked as objects do
    Remove the object marker from P;
    h ← f(P_anc) + λ √(σ̂²_bg + g⁻¹ f(P_anc));
    while f(P) < h do
        if P has a significant descendant then
            P ← mainBranch(P);
        else if P has a descendant then
            P ← mainPowerBranch(P);
        else
            Break.
    Mark P as object;
```

Tree structure. Our method is compared with the segmentation performed by SExtractor 2.19.5. SExtractor settings are kept close to their default values:

- Our background and noise root mean square estimates are used. This already improves the segmentation of SExtractor with respect to the original estimate of SExtractor, that correlates too much with objects.
- `DETECT_MINAREA = 3`. In SExtractor 2.19.5, it represents the minimum number of pixels for a component to be possibly detected as object.
- `FILTER_NAME = default.conv`. It is the default smoothing filter, as seen in Section 4.5.
- `DETECT_THRESH = 1.575`σ above the local background. The default threshold of 1.5 (times the noise standard deviation) is changed to make the expected false positives similar to significance test 4 for noise-only images. Expected false positives per image is approximately 0.38 based on the results of 1000 simulated noise images.

Algorithm 5: `MTObjects`$(I,$ `nodeTest`$, \alpha, g, \lambda)$

Input: Image I, function `nodeTest`, significance level α, gain g, move factor λ.
Result: Max-Tree M. Nodes in M corresponding to objects are marked.

1 $(\hat{\mu}_{\text{bg}}, \hat{\sigma}^2_{\text{bg}}) \leftarrow$ `EstimateBackgroundMeanValueAndVariance`();
2 $I_{i,j} \leftarrow \max(I_{i,j} - \hat{\mu}_{\text{bg}}, 0)$;
3 $M \leftarrow$ create a Max-Tree representation of I;
4 `SignificantNodes`$(M,$ `nodeTest`$, \alpha, g, \hat{\sigma}^2_{\text{bg}})$;
5 `FindObjects`(M);
6 `MoveUp`$(M, \lambda, g, \hat{\sigma}^2_{\text{bg}})$;

– `MEMORY PIXSTACK = 4000000`. To avoid overflows: the value is larger than the number of pixels in an image of our dataset.

While there is no guarantee that these settings are optimal, our comparison gives an impression of the performance of our method. A quantitative comparison on simulated data could be an interesting follow-up to show more precisely the strengths and weaknesses of MTObjects and SExtractor. For the experiments, we used 254 images from the SDSS Data Release 7 [1] catalogue. For every section of the sky, five images are acquired in five different bands of the widely used photometric system (u', g', r', i', z'). We use *r*-band images, because they have the best quality [6].

6.1 Object detection

An object is defined as a lump in the image signal that is not due to noise. All the four significance tests were compared against each other and SExtractor. The significance test 4 returns a larger number of objects in about 100% of the images in the dataset with respect to significance test 1 and 2 and in about 70% with respect to significance test 3 and SExtractor. After inspection of the results, it is clear that, in general, MTObjects preserves more the faint outer structures of objects and nested objects are deblended in a more natural way. Examples can be seen in Fig. 7 and Fig 8. The fainter parts and galactic filaments are identified by MTObjects, for example in Fig. 8e and Fig. 8b. In these two cases the difference is striking. Object deblending by SExtractor does not always work well. Sometimes, weird segmentations appear, such as the one in Fig. 8c. We noticed that MTObjects detects more objects nested in larger objects (galaxies), when the pixel values of the nested objects are above the SExtractor's threshold. For example, Fig. 7b shows a few stars on top of the galaxy segmented as separate objects, whereas in the SExtractor they are for the most part included in the same object as the galaxy, see Fig. 7c. This is explained by the fact that every node in the Max-Tree is used, while SExtractor uses a fixed number of sub-thresholds from its background level to the highest peak component in the object, without considering noise and object properties. To understand if the improved detection of nested objects can explain the better performance of significance test 4, we limited the data set to more compact objects. This is achieved by making a list sorted by area of the largest connected component in each image at the threshold used by SExtractor. The performance of significance test 4 and SExtractor is similar on this new dataset: the difference in the total number of all the objects found in the images is then explained by the number of nested object detections. In practice, in MTObjects it is like if the threshold used by SExtractor is lowered to 0.5, the value of the parameter λ used in `MoveUp`, without increasing the number of false positives: that eases the detection of fainter structures. It is possible to lower λ further, but more noise would be attached to the objects and included in the segmentation.

We tested then how significance test 4 performs in the case of densely spaced overlapping objects. When two identical objects overlap, one of the nodes marked as object has a lower `power` or `powerAlt` value on average. If overlapping objects are close enough to each other and at SExtractor's threshold they are still detected as distinct objects, MTObjects could fail to detect them as separate. A grid filled with small stars is generated with the IRAF software, as in Fig. 9. The magnitude is set to −0.2 to make objects barely detectable

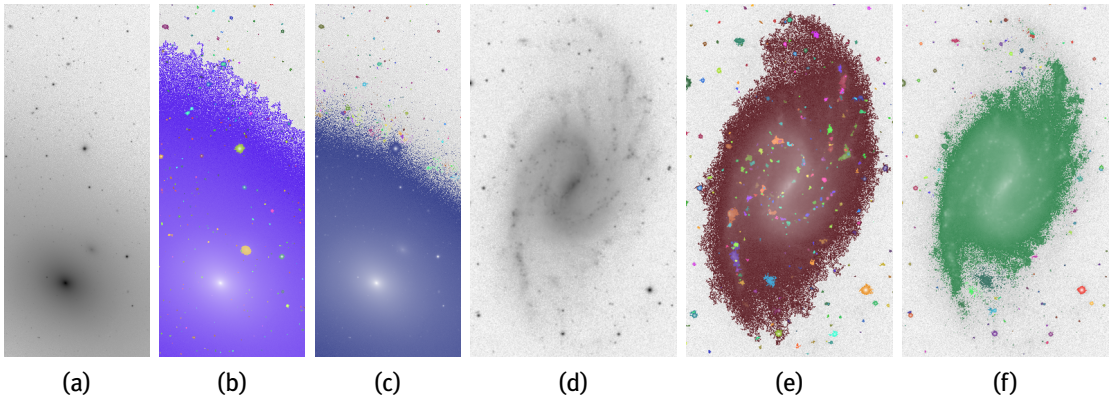


Figure 7: MTObjects identifies better fainter outer regions and the nested objects. Crop of `fpC-003804-r5-0192.fits`: (a) original image; (b) result of significance test 4; (c) result of SExtractor. Crop of `fpC-001332-r4-0066.fits`: (d) original image; (e) result of significance test 4; (f) result of SExtractor.

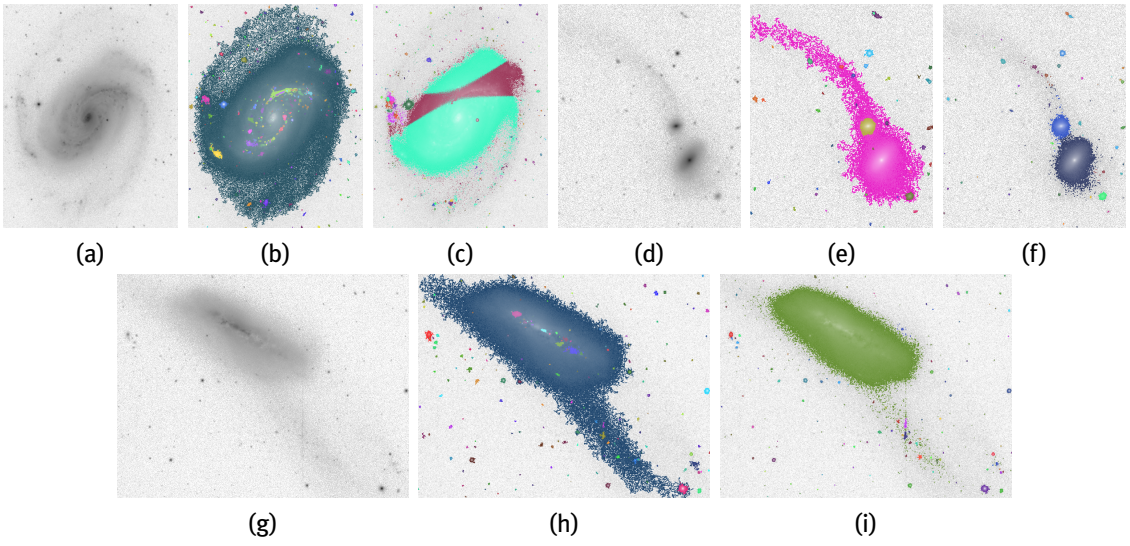


Figure 8: Comparison of objects with faint extended regions. Crop of `fpC-003903-r2-0154.fits`: (a) original image; (b) significance test 4; (c) SExtractor. Crop of `fpC-004576-r2-0245.fits`: (d) original image; (e) significance test 4; (f) SExtracto. Crop of `fpC-004623-r4-0202.fits`: (g) original image; (h) significance test 4; (i) SExtractor.

when noise is added. The diameter of objects is 3 pixels (full width at half maximum). The background equals 1000 at every pixel and the gain is 1. Gaussian noise is added to the image, with the pixel value as mean and variance. In this case of densely spaced objects, SExtractors detects a number of stars closer to the actual number than MTObjects with significance test 4. The results show that a threshold would be better when objects are very densely spaced. We performed a further test with an actual image of a globular cluster. A cluster can be seen as a single object made of a halo caused by the light emitted from a large number of stars close to each other. The fixed threshold of SExtractor seemed to work slightly better. In a globular cluster image that we used, the total number of stellar objects identified by SExtractor is 3164, whereas MTObjects detected 3035. In SExtractor, when the halo is below its fixed threshold, it is considered background. The intensity of stars is estimated relatively to that background. In MTObjects, the estimated intensity of objects is lower, because the halo is not part of the background and considered as a large object. Therefore, some objects have too low intensity to be deblended from the halo region.

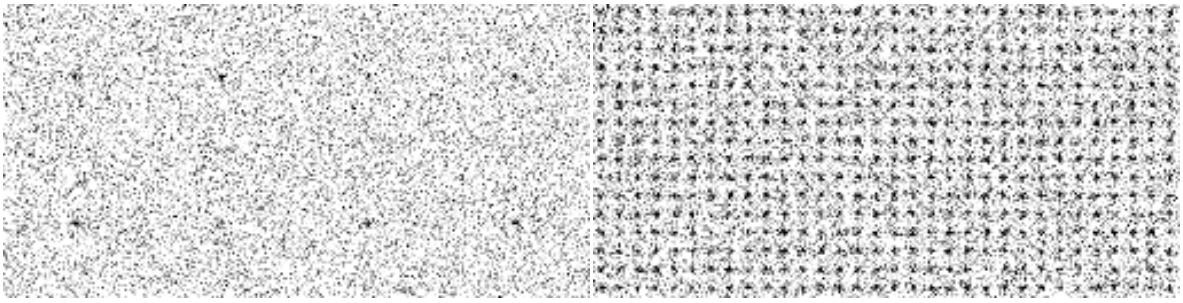


Figure 9: Left image shows part of the sparse stars grid; right image shows part of a dense star grid.

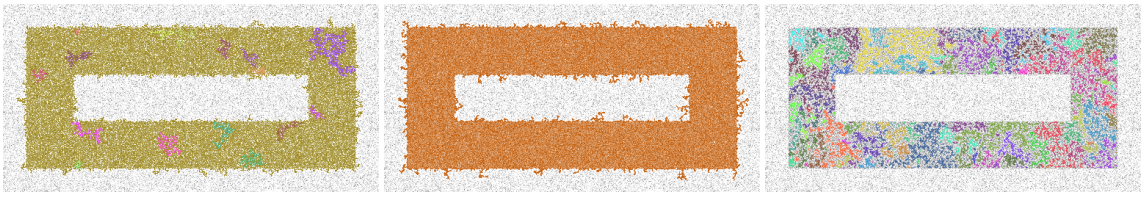


Figure 10: In case of a fragmented simulated object, we show three possible outputs for the significance test 3 (left), significance test 4 (middle) and SExtractor (right). The pixels of the object have the value 1.5, close to the SExtractor's threshold. The background is 0 and Gaussian noise is added with $\sigma = 1$. The image on the right shows a strong fragmentation.

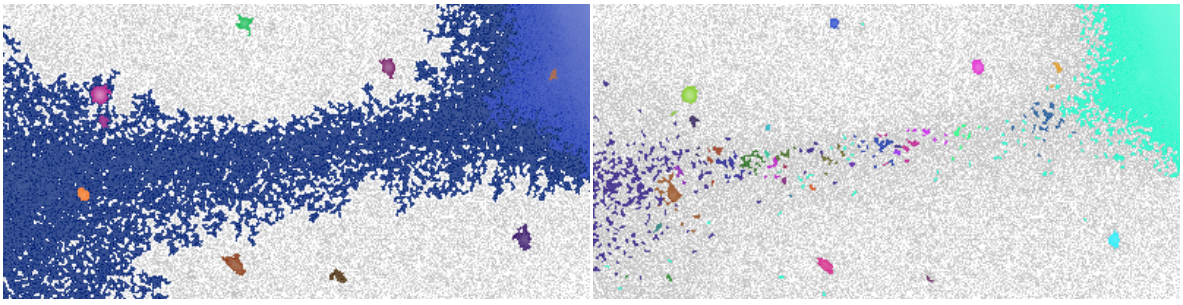


Figure 11: Fragmentation of a thin faint filament between two galaxies, cropped section of file `fpC-002078-r1-0157.fits`. Significance test 4 (left) and SExtractor (right).

6.2 Object fragmentation

A source of false positives, apart from those caused by the statistical tests, is the fragmentation of objects due to noise. An example is shown in Fig. 10. Fragmentation appears to happen in relatively flat structures and the chance is increased if different parts of the structure are thinly connected. If only one pixel connects two parts, the variation in value due to noise can make a deep cut. In the case of the threshold used by SExtractor, fragmentation is severe if the object values are just below the threshold. The expected number of false positives due to fragmentation for the given data set is unknown. Most of the images do not show any evident fragmented objects. An image where it does happen is displayed in Fig. 11, when SExtractor is used. While the SExtractor parameter `CLEAN_PARAM` can be changed to prevent this from happening, it is left to the default as it has a negative effect on the number of objects detected and fragmentation actually happens only for the galaxies in Fig. 11.

6.3 Dust lanes and artifacts

The last possible source of false positive is represented by dust lanes as in Fig. 12 and artefacts as in Fig. 13. In Fig. 12f, the galactic core is split due to a dust line. Fig. 13a, Fig. 13b and Fig. 13c could represent an artefact

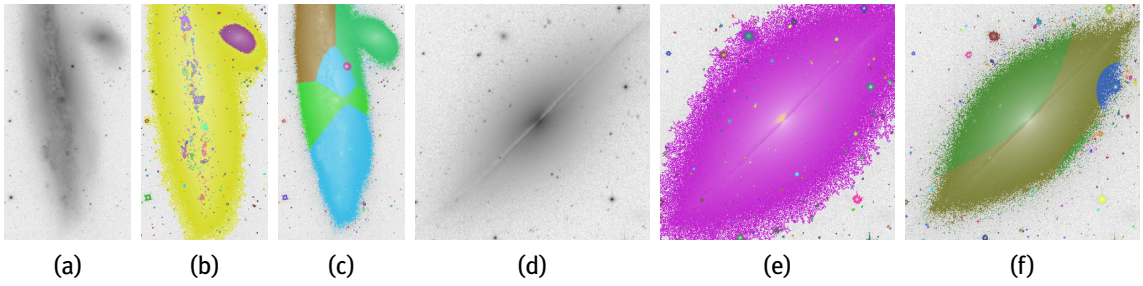


Figure 12: Dust lanes. Crop of `fpC-004623-r4-0202.fits`: (a) original image; (b) significance test 4; (c) SExtractor. Crop of `fpC-001739-r60308.fits`: (d) original image; (e) significance test 4; (f) SExtractor.

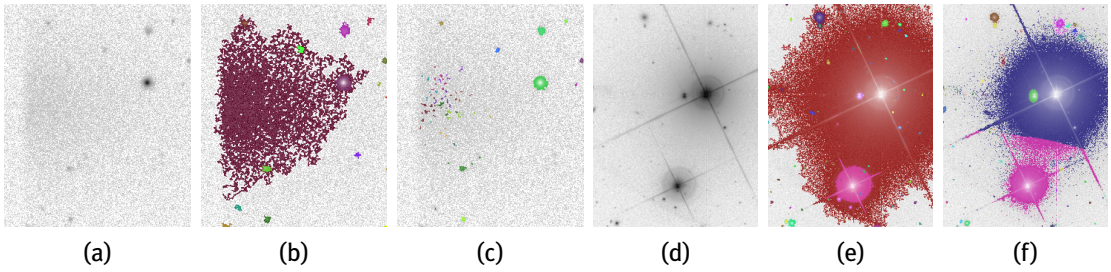


Figure 13: Artefacts. Crop of `fpC-002326-r4-0174.fits`:(a) original image; (b) significance test 4; (c) SExtractor. Crop of `fpC-001345-r3-0182.fits`: (d) original image; (e) significance test 4; (f) SExtractor.

or a vertical cut-off. Refraction spikes, as the one shown in Fig. 13d, can also be a cause of false positives as in the wave-like shape in Fig 13f.

6.4 Experiments on 3D volumes

MTObjects can work also with other datasets, as long as it is possible to provide a suitable noise model to derive an estimate of the noise (background) mean and variance and to study the behaviour of the power attribute, or some other attribute, in the case of noise components. In this section, we want to report a brief summary of the results got in a previous work [9] by the authors and astronomers of the Kapteyn Astronomical Institute of the University of Groningen, The Netherlands. The MTObjects algorithm described in this paper was adapted to identify objects in high resolution 3D volumes containing measurements of the radio spectral line emission of galaxies. A noise model different from the one of the SDSS dataset was used to fit the characteristics of the radio volumes. The negative values in the cube were considered noise. The background estimate is known to be equal to 0 and the variance could be easily estimated using the Median Absolute Deviation method. The background variance does not need to be scaled with the voxel value in the case of radio data. The segmentation of the adapted MTObjects algorithm was compared with the output of SoFiA [15], a source finder used with this kind of data. Specifically, we performed source identification on a 360x360x1464 cube containing the 21 cm neutral hydrogen (HI) emission of galaxies as they would be observed by the WSRT (Westerbork Synthesis Radio Telescope, courtesy of P. Serra). Fig. 14 [9] shows all the objects found in the WSRT cube, applying the significance test 1 described in Section 4.1. The images in Fig. 14 are moment-0 images, computed summing up the flux of the detected sources along the third dimension. For this dataset, MTObjects actually does not identify the faint outer boundaries of the sources and some objects are missed. It was promising, though, that a source identified by MTObjects (circle in Fig. 14b) is missed by the SoFiA source finder. The noise model used turned out not to be ideal. The noise in the WSRT cube is Gaussian but not independent, showing correlations. Therefore, the distribution is not truly χ^2. Experiments showed that the current model does not fit the data very well. In fact, to improve the output of function `SignificantNodes()` in Alg. 2, the `MoveUp()` function in Alg. 4 played an important role: after applying the significance test, many

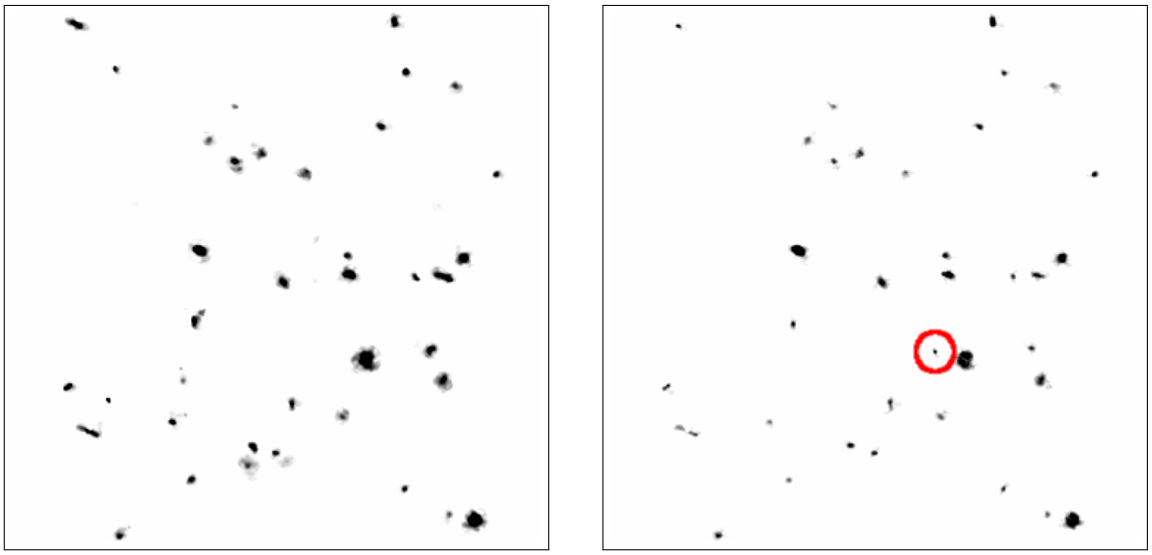


Figure 14: The two images taken from [9] represent moment-0 images of the identified sources in the WSRT cube (a) by SoFiA and (b) by a version of MTObjects tuned to radio volume data. The red circle highlights a source that is not identified by SoFiA.

unwanted nodes were marked as significant and too much noise was attached to the segmented sources. There is currently ongoing work on radio volumes to improve the segmentation, modifying the statistical test, the Max-Tree structure and the attributes chosen.

7 Speed performance

On the SDSS dataset, both for MTObjects and SExtractor, the timer is started before background estimation and is stopped after object classification in SExtractor and after executing `MoveUp` in MTObjects. SExtractor does perform also object classification, far from perfect at lowest magnitudes, classifying objects as stars or galaxies through a neural network approach. The amount of time spent on classification is unknown. MTObjects does not perform any classification. Tests were done on an Intel Core i5-4460 with a single thread. Both methods are quite fast: the median timing on our dataset is 0.7670 seconds for MTObjects and 0.310 for SExtractor. SExtractor is typically 2.5 times faster than MTObjects, if median run-time is considered. When using the mean run-time, SExtractor is 1.3 times faster. SExtractor's execution time is affected by the number of pixels above the fixed threshold: it takes longer time for images that have many pixels above the threshold. MTObjects is more constant in run time and less dependant on the image characteristics. Further optimizations are possible and under study. First tests show that the time of MTObjects can be reduced to approximately the same as SExtractor.

8 Conclusions and future work

The Max-Tree based method (MTObjects) presented in this paper performs better at extracting faint parts of astronomical objects compared to SExtractor, a state-of-the-art method. Our background estimate is less biased by objects than in SExtractor. MTObjects improves the fixed threshold mechanism used by SExtractor by using a statistical test based on the power attribute in each node of the Max-Tree representing the image. The distribution of the power is compared to its expected distribution in case of noise, according to the area of the node. MTObjects is better at extracting faint parts of objects compared to the fixed threshold used by

SExtractor. When an object is defined to have a single maximum pixel value, excluding maxima due to noise, MTObjects is better at finding nested objects. Every possible threshold is tested in MTObjects, whereas SExtractor is bound to a fixed number of thresholds. Deblending objects appears to be better in MTObjects when there is a large difference in size and objects do not have a Gaussian profile. Otherwise, one of the objects will be considered as a smaller branch by MTObjects. A drawback is that too many pixels are assigned arbitrarily to a single object. The SExtractor method of fitting Gaussian profiles makes more sense in this case and allows for a more even split in pixels. This method could be added as post-processing step to MTObjects. MTObjects appears to be slightly worse in case of densely spaced and overlapping objects, like globular clusters.

The power attribute was initially chosen because in the non-filtered case it has a known scaled χ^2 distribution. Better attribute choices could be investigated. Deblending similar sized objects can be improved. Nested significant connected components could in reality represent the same object. The current choice, controlled by λ in `MoveUp` is not ideal. The threshold looks too high for large objects and too low for small objects. Parameter λ could be made variable and dependant on the filter, connectivity and node attributes used. If other noise models are used in other data sets, background mean and variance estimates, and significance tests can be adjusted accordingly. The degree of smoothing applied helps to avoid fragmentation and it should be further investigated. Currently, the rejection boundaries are approximated by simulations which must be recomputed for every filter and significance level. Knowing the exact distributions will speed up this phase, but it is often not feasible.

Acknowledgement: This work was funded by the Netherlands Organisation for Scientific Research (NWO) under project number 612.001.110.

References

[1] K. N. Abazajian et al., Astrophys. J. Suppl. S., 2009, 182, 2
[2] C. Berger, T. Géraud, R. Levillain, N. Widynski, A. Baillard, E. Bertin, Proceedings of International Conference on Image Processing, Sep. 16-19, 2007, San Antonio, TX, USA 41
[3] E. Bertin, S. Arnouts, Astron. Astrophys. Suppl. S., 1996, 117, 393
[4] E. Breen, R. Jones, Comput. Vis. Image. Und., 1996, 64, 3
[5] R. B. D'Agostino, A. Belanger, R. B. D'Agostino Jr, Am. Stat., 1990, 44, 4
[6] J. Gunn et al., Astron. J., 1998, 116, 6
[7] M. Masias, J. Freixenet, X. Lladó, M. Peracaula, Mon. Not. R. Astron. Soc., 2012 422, 1674
[8] M. Masias, M. Peracaula, J. Freixenet, X. Lladó: Exp. Astron., 2013, 36, 591
[9] U. Moschini et al., Proceedings of the 2014 conference on Big Data from Space (BiDS'14), Nov. 12-14, 2014, Frascati, Italy (Publications Office of the European Union) 232
[10] G. K. Ouzounis, M. H. F. Wilkinson, IEEE T. Pattern Anal., 2011, 33, 224
[11] B. Perret, S. Lefevre, C. Collet, E. Slezak, Proceedings of International Conference on Pattern Recognition, Aug. 23-26, 2010, Istanbul, Turkey (IEEE) 4089
[12] P. Salembier, A. Oliveras, L. Garrido, IEEE T. Image Process., 1998, 7, 555
[13] sdss.org, 2007: Photometric flux calibration - published online: http://www.sdss2.org/dr7/algorithms/fluxcal.html
[14] J. Serra, Image Analysis and Mathematical Morphology, Part II: Theoretical Advances, 1988, Academic Press, London
[15] P. Serra et al., Mon. Not. R. Astron. Soc., 2015, 448, 2
[16] C. Stoughton et al., Astron. J., 2002, 123, 1 (2002)
[17] P. Teeninga, U. Moschini, S. C. Trager, and M. H. F. Wilkinson, 11th International Conference Pattern Recognition and Image Analysis: New Information Technologies, Sep. 23-28, 2013, Samara, Russia (IPSI RAS) 746
[18] P. Teeninga, U. Moschini, S. C. Trager, M. H. F. Wilkinson, Proceedings of International Conference on Image Processing, Sep. 27-30, 2015, Quebec City, Canada, (IEEE)
[19] N. Young, A. N. Evans, IEE P-Vis. Image Sign., 2003, 150, 5